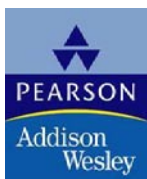


5.2

List Boxes

List Boxes Display a List of Items and Allow the User to Select an Item From the List





ListBox Items Property

- This property holds the entire *list* of items from which the user may choose
- The list values may be established at run time or as part of the form design
- To set list values in the form design:
 - Select the list box in the *Design* window
 - Click on ellipses button for the *Items property*
 - This property is a *collection*, a list of values
 - Type each value on a separate line



ListBox Items.Count Property

- This property holds the number of items that are stored in the Items property

- Example of use:

```
If lstEmployees.Items.Count = 0 Then  
    MessageBox.Show("The list has no items!")  
End If
```

- The number of items in the list can be assigned to an integer variable

```
numEmployees = lstEmployees.Items.Count
```



Item Indexing

- The Items property values can be accessed from your VB code
- Each item value is given a sequential index
 - The first item has an index of 0
 - The second item has an index of 1, etc.
- Example:

```
name = lstCustomers.Items(2)
' Access the 3rd item value
```



Index Out of Range Error

- The index of the last item is always `list.Items.Count-1`
- Reference to an index greater than `Count-1` or less than zero throws an exception
- An exception handler can trap this error
- The variable `ex` captures the exception thrown

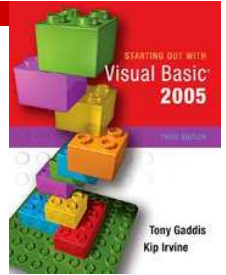
Try

```
strInput = lstMonths.Items(n).ToString()
```

Catch ex as Exception

```
MessageBox.Show(ex.Message)
```

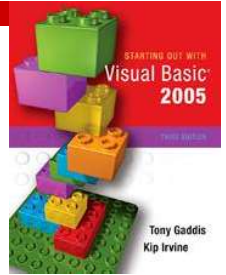
End Try



ListBox SelectIndex Property

- Use the *SelectIndex property* to retrieve the index of an item selected by the user
- If no item is selected, the value is set to -1 (an invalid index value)
- Can use SelectIndex to determine if an item has been selected by comparing to -1
- Example:

```
If lstLocations.SelectedIndex <> -1 Then  
    location = lstLocations.Items(lstLocations.SelectedIndex)  
End If
```



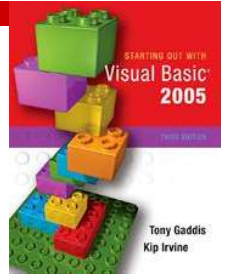
ListBox SelectedItem Property

- Instead of using the SelectedIndex property as follows:

```
If lstMonths.SelectedIndex <> -1 Then  
    month = lstMonths.Items(lstMonths.SelectedIndex)  
End If
```

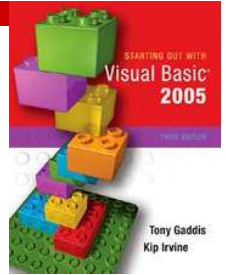
- The *SelectedItem property* can be used to retrieve a selected item value as shown

```
If lstMonths.SelectedIndex <> -1 Then  
    month = lstMonths.SelectedItem.Value)  
End If
```



ListBox Sorted Property

- *Sorted* is a boolean property
- If true, causes the values in the Items property to be displayed in alphabetical order
- If false, values in the Items property are displayed in the order they were added



ListBox.Items.Add Method

- Items can be added to the end of a ListBox list in your VB code using the *Add method*
- Format is
`ListBox.Items.Add(Item)`
- *ListBox* is the name of the control
- *Item* is the value to add to the Items property
- Example:
`lstStudents.Items.Add("Sharon")`



ListBox.Items.Insert Method

- Items can be added at a specific position of a ListBox in VB code using the *Insert method*
`ListBox.Items.Insert(Index, Item)`
- *Index* specifies position where *Item* is placed
- *Index* is zero based similar to SelectedIndex property
- Items that follow are “pushed” down
- Example inserting "Jean" as the 3rd item
`1stStudents.Items.Insert(2, "Jean")`



ListBox Methods to Remove Items

- ***ListBox.Items.RemoveAt (Index)***
 - Removes item at the specified index
- ***ListBox.Items.Remove (Item)***
 - Removes item with value specified by *Item*
- ***ListBox.Items.Clear ()***
 - Removes all items in the Items property
- Examples:

```
1stStudents.Items.RemoveAt (2)
```

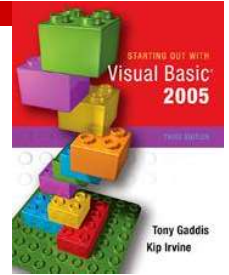
```
`remove 3rd item
```

```
1stStudents.Items.Remove ("Jean")
```

```
`remove item Jean
```

```
1stStudents.Items.Clear ()
```

```
`remove all items
```



Other ListBox Methods

- ***ListBox.Items.Contains (Item)***
 - Returns true if *Item* is found in the collection
- ***ListBox.Items.IndexOf (Item)***
 - Returns an integer with the index position of the first occurrence of *Item* in the collection
- Examples:

```
blnFound = lstMonths.Items.Contains ("March")  
intIndex = lstMonths.Items.IndexOf ("March")
```
- Tutorial 5-1 provides more examples of ListBox controls, methods and properties